HP 8719C, 8720C, and 8722A/C network analyzers

BASIC Programming Guide

for use with HP 9000 series 200/300 computers

HEWLETT PACKARD

HP Part No. 08720-90156
Printed in USA September 1992

Contents

Figures

Tables

1

Programming Basics

Introduction

This manual is an introduction to remote operation of the HP 8719C, HP 8720C, or HP 8722A/C Network Analyzer using an HP 9000 series 200 or 300 computer. It is a tutorial introduction, using BASIC programming examples.

The reader should become familiar with the operation of the network analyzer before controlling it over HP-IB. This manual is not intended to teach BASIC programming or to discuss HP-IB theory; refer to the following documents which are better suited to these tasks.

- For more information concerning the operation of the network analyzer, refer to the following:

  HP 8719C/8720C User's Guide

  HP 8719C/8720C Reference Manual

- For more information concerning BASIC, refer to the manual set for the BASIC revision being used:

  BASIC Programming Techniques

  BASIC Language Reference

- For more information concerning HP-IB, refer to the following:

  HP-IB Programming Reference

  BASIC Interfacing Techniques

  Tutorial Description of the Hewlett-Packard Interface Bus

  Condensed Description of the Hewlett-Packard Interface Bus

Start-up

Required Equipment

1. HP 8719C, HP 8720C, or HP 8722A/C Network Analyzer
2. HP 9000 Series 200 or 300 computer with enough memory to hold BASIC, needed binaries (refer to "Powering Up the System"), and at least 64 kilobytes of program space.

   A disk drive is required to load BASIC, if no internal disk drive is available.
3. BASIC 3.0 or higher operating system.
4. HP 10833A/B/C/D HP-IB cables to interconnect the computer, the network analyzer, and any peripherals.

Optional Equipment

1. HP 85053B 3.5 mm calibration kit
2. HP 85132C Cable
3. Accessory kit
4. Device under test (DUT)
5. Cables to connect DUT
6. Printer

Powering Up the System

1. Set up the network analyzer as shown in Figure 1-1.

   Connect the network analyzer to the computer with an HP-IB cable.

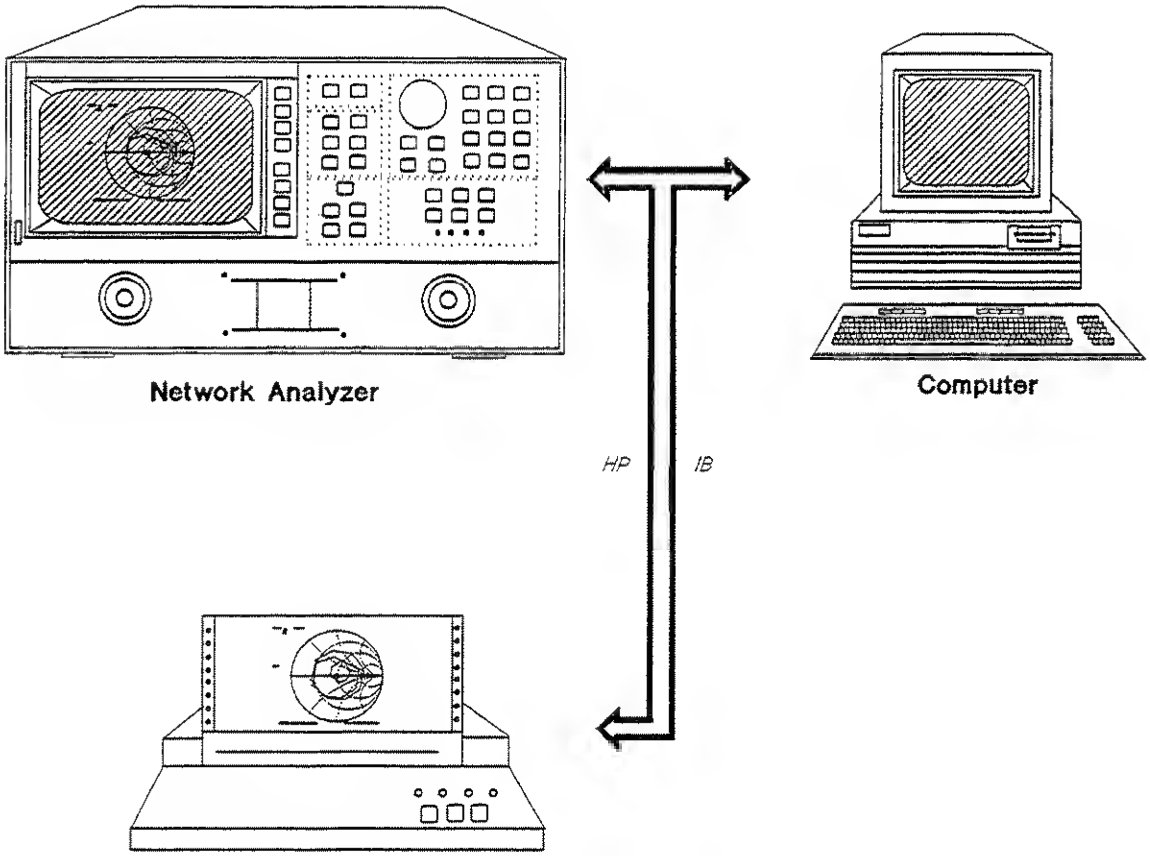

Figure 1-1. HP-IB Connections in a Typical Setup

2. Turn on the computer and load the BASIC operating system.

   Load the following BASIC binary extensions:

   HPIB, GRAPH, IO, KBD, and ERR.

   Depending on the disk drive, a binary such as CS80 may be required.

3. Turn the network analyzer ON.

   To verify the network analyzer's address, press (LOCAL) and select SET ADDRESSES ADDRESS: INSTRUMENT. If the address has been changed from the default value (16), return it to 16 while performing the examples in this document by pressing (1) (6) (x1) and the presetting the network analyzer.

   Make sure the network analyzer is in the TALKER/LISTENER or USE PASS CONTROL mode, as indicated under the (LOCAL) key. These are the only modes in which the network analyzer will accept HP-IB commands.

4. On the computer, type the following:

   OUTPUT 716;"PRES;" (Return) (or (EXECUTE))

This will preset the network analyzer. If preset does not occur, there is a problem. First check all HP-IB addresses and connections: most HP-IB problems are caused by an incorrect address and bad or loose HP-IB cables.

Note

Only the HP 9000 Model 226 and 236 computers have an (EXECUTE) key. The Model 216 has an (EXEC) key with the same function. All other computer use the (Return) key as both execute and enter. The notation (Return) is used in this document.

Measurement Programming

This section describes how to organize the commands into a measurement sequence. Figure 1-2 shows a typical measurement sequence.

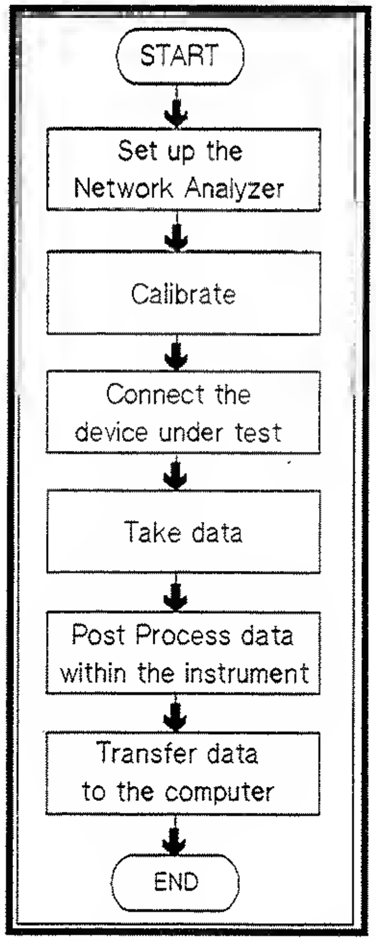

Figure 1-2. Typical Measurement Sequence

- Setting up the network analyzer

  Define the measurement by setting all of the basic measurement parameters. These include all the stimulus parameters: sweep type, span, sweep time, number of points, and RF power level. They also include the parameter to be measured, and both IF averaging and IF bandwidth. These parameters define the way data is gathered and processed within the instrument, and to change one requires that a new sweep be triggered.

  There are other parameters that can be set within the network analyzer that do not affect data gathering directly, such as smoothing, trace scaling or trace math. These functions are classed as post processing functions: they can be changed with the network analyzer in the hold mode, and the data will correctly reflect the current state.

  The save/recall registers provide a rapid way of setting up an entire instrument state.

- Calibrating

Perform a measurement calibration once the network analyzer state has been defined. Measurement calibration is not required to make a measurement, but is highly recommend with microwave network analyzers in order to improve measurement accuracy.

There are several ways to calibrate the network analyzer as follows:

- Pause or stop the program and have the operator perform the calibration using the front panel keys.
- Guide the operator through the calibration under computer control, as discussed in "Frequency Response Calibration" in Chapter 2 and "1-Port Reflection Calibration" in Chapter 2.
- Transfer calibration data from a previous calibration back into the instrument, as discussed in "Reading Calibration Data" in Chapter 3.

- Connecting device under test

  Connect and adjust the device under test (DUT). The adjustment process can be assisted by setting specific network analyzer functions, such as limit testing, bandwidth searches, and trace statistics. All adjnstments to the DUT should be performed at this stage.

- Taking data

  With the device connected and adjusted, command the network analyzer to take the measurement data for subsequent transfer.

  The single sweep command `SING` is designed to ensure a valid sweep. When the sweep is complete, the network analyzer is put into the hold mode, storing the data inside the network analyzer. The number of groups command `NUMGn` is designed to work the same as single sweep, except that it triggers n sweeps. This is nseful when making a measurement with an averaging factor of n. Both single sweep and number of groups restart averaging.

- Post-processing

  With valid data to operate on, utilize the post-processing functions for analyzing the data. Referring to Figure 1-3, any function that affects the data after the error correction stage can be used. Some useful functions are trace statistics, marker searches, and electrical delay. If a 2-port calibration is active, then any of the fonr S-parameters can be viewed without taking a new sweep.

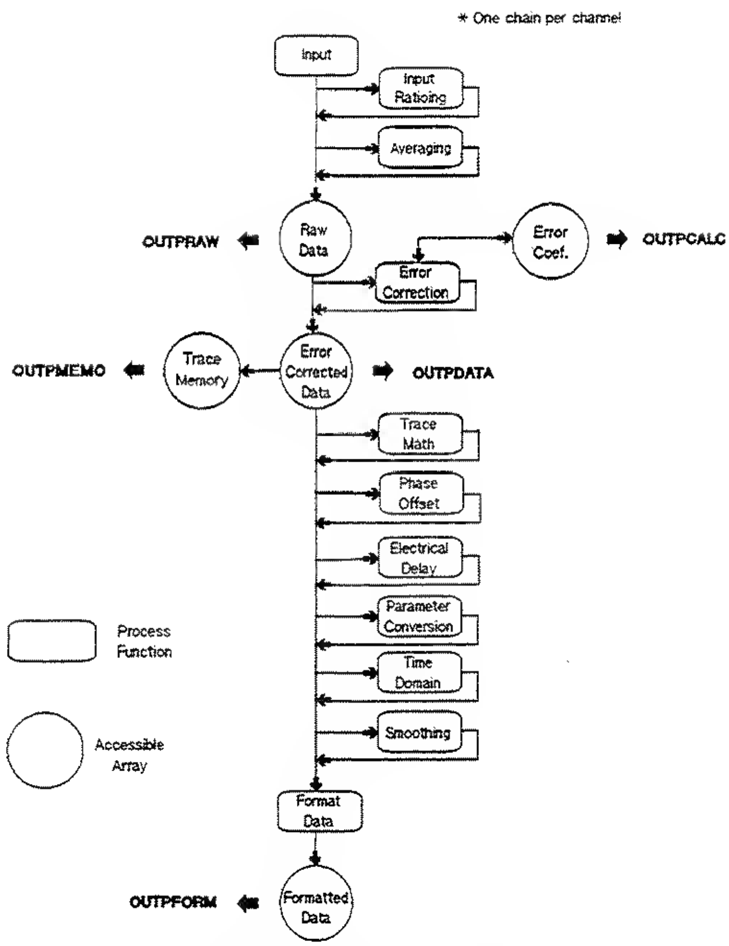

Figure 1-3. Data Processing Chain

- Transferring data

  Lastly, read the results out of the network analyzer. All the data output commands are designed to ensure that the data transmitted reflects the current state of the network analyzer:

  - ☐ OUTPDATA, OUTPRAWn, OUTPFORM for transfer of entire data traces.
  - ☐ OUTPLIML, OUTPLIMM, and OUTPLIMF for transfer of limit testing results.
  - ☐ OUTPMARK for transfer of the currently active marker's results. This command will activate a marker if one is not already selected.
  - ☐ OUTPMSTA for transfer of statistics that have been calculated for data between the active marker and the delta reference marker. If there is no delta reference, the entire trace data is used.
  - ☐ OUTPMWID for transfer of the results of a bandwidth search.

  Data transfer is discussed further in "Data Transfer from the Network Analyzer to a Computer" in Chapter 2.

2

Basic Programming Examples

Setting Up a Measurement

In general, the procedure for setting up measurements on the network analyzer via HP-IB follows the same sequence as if the setup was performed manually. There is no required order, as long as the desired frequency range, number of points and power level are set prior to performing the calibration.

By interrogating the network analyzer to determine the actual values of the start, the stop, or the center frequencies, or the frequency span, the computer can keep track of the actual frequencies.

This example illustrates how a basic measurement can be set up on the network analyzer. The program will first select the desired parameter, the measurement format, and then the frequency range.

This example sets up a measurement of transmission log magnitude on channel 1. When prompted for the center frequency and the frequency span, enter any value in Hz from 1.0E+5 (for the S-parameter Test Set) to 5.0E+8. These will be entered into the network analyzer, and the frequencies will be displayed.

```
10    !
20    ! Setting Up a Measurement
30    !
40    Hp8720=716
50    ABORT 7
60    CLEAR Hp8720
70    !
80    OUTPUT Hp8720;"PRES;"
90    OUTPUT Hp8720;"CHAN1; S21; LOGM"
100   INPUT "Enter center frequency (Hz)",F_cent
110   INPUT "Enter frequency span (Hz)",F_span
120   OUTPUT Hp8720;"CENT ";F_cent
130   OUTPUT Hp8720;"SPAN ";F_span
140   !
150   OUTPUT Hp8720;"CENT?"
160   ENTER Hp8720;F_centr
170   OUTPUT Hp8720;"SPAN?"
180   ENTER Hp8720;F_spanr
190   PRINT "Center frequency:",F_centr;"Hz"
200   PRINT "Frequency span:",F_spanr;"Hz"
210   END
```

Figure 2-1. Sample Program: Setting Up a Measurement

Program explanation

Line 40	Assigns network analyzer HP-IB address.
Lines 50 and 60	Prepares for HP-IB control.
Line 80	Presets the network analyzer.
Line 90	Makes channel 1 the active channel, and measures the transmission parameter, S_{21}, displaying its magnitude in dB.
Lines 100 and 110	Inputs the center frequency and the frequency span.
Lines 120 and 130	Sets the center frequency and the frequency span.
Lines 150 through 180	Queries the center frequency and the frequency span.
Lines 190 and 200	Shows the current center frequency and the frequency span.

Performing a Measurement Calibration

This section will demonstrate how to coordinate a measurement calibration over HP-IB. The HP-IB program follows the key strokes required to calibrate from the front panel: there is a command for every step.

The general keystrokes sequence is to select the calibration, measure the calibration standards, and then declare the calibration done. The actual sequence depends on the calibration kit and the calibration type.

Calibration Kits

The calibration kit tells the network analyzer what standards to expect at each step of the calibration. The set of standards associated with a given calibration is termed a class. Refer to the *HP-IB Programming Reference* for the relation between the calibration types and the standard classes.

For example, measuring the SHORT during a 1-port calibration is one calibration step. All of the SHORTs that can be used for this calibration step make up the class, which is called class S_{11}B. For the 7 mm calibration kits, class S_{11}B has only one standard in it. For the Type N calibration kit, class S_{11}B has two standards in it: male and female SHORTs.

When doing a 1-port calibration in 7 mm over HP-IB, sending `CLASS11B` will automatically measnre the SHORT. In Type N, sending `CLASS11B` brings up the menu with the male and female SHORT options. To select a standard, use `STANA` or `STANB`. The `STAN` command is appended with the letters A through G, corresponding to the standards list under softkeys 1 through 7, softkey 1 being the topmost softkey.

Each full 2-port calibration is divided into three sub-sequences: transmission, reflection, and isolation. Each sub-sequence is treated like a calibration in its own right; each must be opened, have all the standards measured, and then be declared done. The opening and closing statements for the transmission sub-sequence are `TRAN` and `TRAD`. The opening and closing statements for the reflection sub-sequence are `REFL` and `REFD`. The opening and closing statements for isolation are `ISOL` and `ISOD`.

Frequency Response Calibration

The following program does a response calibration using a THRU calibration device. This program simplifies the calibration for the operator by giving explicit directions on the computer's display.

```
10    !
20    ! Frequency Response Calibration
30    !
40    Hp8720=716
50    ABORT 7
60    CLEAR Hp8720
70    !
80    OUTPUT Hp8720;"PRES;"
90    OUTPUT Hp8720;"CHAN1; S21; LOGM;"
100   INPUT "Enter center frequency (Hz)",F_cent
110   INPUT "Enter frequency span (Hz)",F_span
120   OUTPUT Hp8720;"CENT";F_cent
130   OUTPUT Hp8720;"SPAN";F_span
140   !
150   OUTPUT Hp8720;"HOLD;"
160   OUTPUT Hp8720;"CALK35MM;"
170   OUTPUT Hp8720;"CALIRESP;"
180   INPUT "Connect THRU, then press [Return].",Dum$
190   OUTPUT Hp8720;"CLES;"
200   OUTPUT Hp8720;"STANC;"
210   REPEAT
220     OUTPUT Hp8720;"ESB?;"
230     ENTER Hp8720;Stat
240   UNTIL BIT(Stat,0)
250   !
260   OUTPUT Hp8720;"*OPC?;RESPDONE;"
270   ENTER Hp8720;Dum
280   OUTPUT Hp8720;"CONT;"
290   DISP "Response cal completed."
300   END
```

Figure 2-2. Sample Program: Frequency Response Calibration

Program explanation

Line 150	Sets the trigger to the hold mode.
Line 160	Selects the 3.5 mm calibration kit.
Line 170	Opens the calibration by calling the response calibration.
Line 180	Asks for a THRU, and waits for the operator to connect it.
Line 190	Clears all status registers.

Line 200	Selects and measures the THRU. There is more than one standard in this calibration, so we must identify the specific standard within this calibration. The THRU is the third softkey selection from the top in the menu, so use the `STANC` command to select THRU as the standard.
Lines 210 through 240	Waits for the standard to be measured. This is indicated by bit 0 of event status register B.
Lines 260 through 270	Affirms the completion of the calibration, and waits for calculation completion.
Line 280	Sets the trigger to the continuous mode.

1-Port Reflection Calibration

The following program does a 1-port calibration using the HP 85052D 3.5 mm calibration kit. This program simplifies the calibration for the operator by giving explicit directions on the computer display.

```
10    !
20    ! 1-port Reflection Calibration
30    !
40    Hp8720=716
50    ABORT 7
60    CLEAR Hp8720
70    !
80    OUTPUT Hp8720;"PRES"
90    OUTPUT Hp8720;"CHAN1;"
100   INPUT "Enter center frequency (Hz)",F_cent
110   INPUT "Enter frequency span (Hz)",F_span
120   OUTPUT Hp8720;"CENT ";F_cent
130   OUTPUT Hp8720;"SPAN ";F_span
140   !
150   OUTPUT Hp8720;"HOLD;"
160   OUTPUT Hp8720;"CALK35MM;"
170   OUTPUT Hp8720;"CALIS111;"
180   !
190   INPUT "Connect OPEN at port 1, then press [Return].",Dum$
200   OUTPUT Hp8720;"OPC;CLASS11A;"
210   GOSUB Op_end
220   !
230   INPUT "Connect SHORT at port 1, then press [Return].",Dum$
240   OUTPUT Hp8720;"OPC;CLASS11B;"
250   GOSUB Op_end
260   !
270   INPUT "Connect LOAD at port 1, then press [Return].",Dum$
280   OUTPUT Hp8720;"CLASS11C;OPC;STANA;"
290   GOSUB Op_end
300   OUTPUT Hp8720;"DONE;"
310   !
320   OUTPUT Hp8720;"OPC?;"
330   OUTPUT Hp8720;"SAV1;"
340   ENTER Hp8720;Dum
350   OUTPUT Hp8720;"CONT"
360   DISP "1-port cal completed."
370   STOP
380   !
390 Op_end: !
400   REPEAT
410     OUTPUT Hp8720;"ESB?"
420     ENTER Hp8720;Stat
430   UNTIL BIT(Stat,0)
440   RETURN
```

```
450   END
```

Figure 2-3. Sample Program: 1-port Reflection Calibration

Program explanation

Line 170	Opens the calibration by calling the S_{11} 1-port calibration.
Line 200 through 210	Selects the OPEN (S11A) class. Since there is only one standard in this class, only the class command needs to be sent. OPC (Operation Complete) along with the subroutine Op_end causes the program to wait under the measurement is completed.
Line 240 through 250	Selects the SHORT (S11B) class. Since there is only one standard in this class, only the class command needs to be sent. OPC (Operation Complete) along with the subroutine Op_end causes the program to wait under the measurement is completed.
Line 280	Selects the LOADS (S11C) class, followed by the BROADBAND load standard, and starts measuring the standard.
Line 320 through 330	Saves the calibration.
Line 350	Sets the trigger to the continuous mode.
Line 390 through 440	Waits until the operation complete bit of the event status register is set to 0.

Full 2-Port Measurement Calibration

The following example shows how to perform a full 2-port measurement calibration using the HP 85052D calibration kit. The main difference between this example and the preceding is that in this case, the calibration porcess allows removal of both the forward and reverse error terms, so that all four S-parameters of the device under test can be measured.

```
1      !
2      ! Full 2-port measurement calibration.
3      ! It guides the operator through a full 2-port calibration,
4      ! using the HP 85052D 3.5 mm economy calibration kit (no sliding loads).
5      !
8      Hp8720=716
10     ABORT 7
20     CLEAR Hp8720
30     OUTPUT Hp8720;"CALK35MM;MENUOFF;"
40     OUTPUT Hp8720;"CALIFUL2;"
50     OUTPUT Hp8720;"REFL;"
60     INPUT "CONNECT OPEN AT PORT 1",Dum$
70     OUTPUT Hp8720;"OPC?;CLASS11A;"
80     ENTER Hp8720;Reply
90     INPUT"CONNECT SHORT AT PORT 1",Dum$
100    OUTPUT Hp8720;"OPC?;CLASS11B;"
110    ENTER Hp8720;Reply
120    INPUT"CONNECT BROADBAND LOAD AT PORT 1",Dum$
130    OUTPUT Hp8720;"CLASS11C;OPC?;STANA;"
140    ENTER Hp8720;Reply
150    INPUT"CONNECT OPEN AT PORT 2",Dum$
160    OUTPUT Hp8720;"OPC?;CLASS22A;"
170    ENTER Hp8720;Reply
180    INPUT"CONNECT SHORT AT PORT 2",Dum$
190    OUTPUT Hp8720;"OPC?;CLASS22B;"
200    ENTER Hp8720;Reply
210    INPUT"CONNECT BROADBAND LOAD AT PORT 2",Dum$
220    OUTPUT Hp8720;"CLASS22C;OPC?;STANA;"
230    ENTER Hp8720;Reply
240    OUTPUT Hp8720;"REFD;"
250    DISP "COMPUTING REFLECTION CALIBRATION COEFFICIENTS"
260    OUTPUT Hp8720;"TRAN;"
270    INPUT"CONNECT THRU [PORT1 TO PORT 2]",Dum$
280    DISP "MEASURING FORWARD TRANSMISSION"
290    OUTPUT Hp8720;"OPC?;FWDT;"
300    ENTER Hp8720;Reply
310    OUTPUT Hp8720;"OPC?;FWDM;"
320    ENTER Hp8720;Reply
330    DISP "MEASURING REVERSE TRANSMISSION"
340    OUTPUT Hp8720;"OPC?;REVT;"
350    ENTER Hp8720;Reply
360    OUTPUT Hp8720;"OPC?;REVM;"
370    ENTER Hp8720;Reply
```

```
380   OUTPUT Hp8720;"TRAD"
390   INPUT"ISOLATE TEST PORTS",Dum$
400   OUTPUT Hp8720;"ISOL;"
410  !OUTPUT Hp8720;"OMII;"                         !IF ISOLATION CAL NOT DESIRED
420  !GOTO 580                                      !SKIP ISOLATION CAL SEQUENCE
430   DISP "MEASURING REVERSE ISOLATION"
440   OUTPUT Hp8720;"OPC?;REVI;"
450   ENTER Hp8720;Reply
460   DISP "MEASURING FORWARD ISOLATION"
470   OUTPUT Hp8720;"OPC?;FWDI;"
480   ENTER Hp8720;Reply
490   OUTPUT Hp8720;"ISOD;"
510   DISP "COMPUTING CALIBRATION COEFFICIENTS"
520   OUTPUT Hp8720;"OPC?;SAV2;"
530   ENTER Hp8720;Reply
540   DISP "DONE"
550   OUTPUT Hp8720;"MENUON;"
560   END
```

Figure 2-4. Sample Program: Full 2-port Measurement Calibration

Program explanation

Line 30	Begin by selecting the 3.5 mm cal kit and turning off the softkey menu.
Line 40	Open the calibration by calling for a full 2-port calibration type.
Line 50	Open the reflection calibration subsequence.
Line 60	Prompts for the OPEN connection and waits for an input to continue.
Line 70 and 80	Use operation complete for measurement of the S11A class.
Line 90	Prompts for the SHORT connection and waits for an input to continue.
Line 100 and 110	Use operation complete for measurement of the S11B class.
Line 120	Prompts for the BROADBAND LOAD connection and waits for an input to continue.
Line 130 and 140	Use operation complete for measurement of standard "A" in the S11C class.
Line 150 through 230	Measure port 2 reflection standards.
Line 240 and 250	Complete the reflection calibration subsequence.
Line 260 and 270	Open the transmission calibration snbsequence.
Line 280 through 370	Measure the four transmission classes.
Line 380	Complete the transmission calibration subsequence.
Line 390 and 400	Open the isolation calibration subsequence.

Line 410 through 490	Measure the two isolation classes.
Line 510 through 550	Finish up by saving the error coefficient arrays and turning softkey menuing on.

Data Transfer from the Network Analyzer to a Computer

Trace information can be read out of the network analyzer in several ways. Data can be read off the trace selectively using the markers, or the entire trace can be read out.

Using Markers to Obtain Trace Data at Specific Points

If only specific information such as a single point off the trace or the result of a marker search is needed, the marker output command can be used to read the information.

Marker data is read out with the command OUTPMARK. This command causes the network analyzer to transmit three numbers: marker value 1, marker value 2, and marker stimulus value. Refer to Table 2-1 for all the different possibilities for values one and two.

```
10   !
20   ! Using Markers to Obtain trace data at specific points
30   !
40   Hp8720=716
50   ABORT 7
60   CLEAR Hp8720
70   !
80   OUTPUT Hp8720;"PRES;"
90   OUTPUT Hp8720;"CHAN1; S21; LOGM;"
100  INPUT "Enter center frequency (Hz)",F_cent
110  INPUT "Enter frequency span (Hz)",F_span
120  OUTPUT Hp8720;"CENT ";F_cent
130  OUTPUT Hp8720;"SPAN ";F_span
140  !
150  OUTPUT Hp8720;"OPC;"
160  OUTPUT Hp8720;"SING;"
170  REPEAT
180    OUTPUT Hp8720;"ESB?"
190    ENTER Hp8720;Stat
200  UNTIL BIT(Stat,0)
210  !
220  OUTPUT Hp8720;"AUTO;"
230  OUTPUT Hp8720;"MARK1;"
240  OUTPUT Hp8720;"SEAMIN;"
250  OUTPUT Hp8720;"OUTPMARK;"
260  ENTER Hp8720;Val1,Val2,Stim
270  PRINT "Min val:",Val1;"dB"
280  PRINT "Stimulus:",Stim;"Hz"
290  END
```

Figure 2-5. Sample Program: Using Markers to Obtain Trace Data at Specific Points

Program explanation

Lines 150 through 200	Collects one sweep of data, and wait for completion.
Line 220	Brings the trace data in view on the network analyzer's display.
Line 230	Activates marker 1.
Line 240	Search for the trace minimum.
Line 250	Outputs the marker values at that point.
Line 260	Reads marker value 1, marker value 2, and the stimulus value. In log magnitude format, the marker value 2 is not significant, but is included for contistency with all data transfers.

Table 2-1. Units as a Function of Display Format

Display Format	Marker Mode	OUTPMARK Marker Readout[1] value 1, value 2	OUTPFORM value 1, value 2
LOG MAG		dB,[2]	dB,[2]
PHASE		degrees,[2]	degrees,[2]
DELAY		seconds,[2]	seconds,[2]
SMITH	LIN MKR	lin mag, degrees	real, imag
CHART	LOG MKR	dB, degrees	real, imag
	Re/Im	real, imag	real, imag
	R + jX	real, imag ohms	real, imag
	G + jB	real, imag Siemens	real, imag
POLAR	LIN MKR	lin mag, degrees	real, imag
	LOG MKR	dB, degrees	real, imag
	Re/Im	real, imag	real, imag
LIN MAG		lin mag,[2]	lin mag,[2]
REAL		real,[2]	real,[2]
SWR		SWR,[2]	SWR,[2]

1 The marker readout values are the marker values displayed in the upper right hand corner of the display. They also correspond to the value and aux value associated with the fixed marker.

2 Value 2 not significant in this form, but is included in data transfers.

Trace Transfer

Getting trace data out of the network analyzer with a 200/300 series computer can be broken down into three steps:

1. Setting up the receive array.
2. Telling the network analyzer to transmit the data.
3. Accepting the transferred data.

Data inside the network analyzer is always stored in pairs, to accommodate real/imginary values, for each data point. Therefore, the receiving array has to be two elements wide, and as deep as the number of points. This memory space must be allocated in the computer (through a DIMension or ALLOCATE statement).

Data Format

The network analyzer can transmit data over HP-IB in five different formats. The type of format affects what kind of data array is declared (real or integer), since the format determines what type of data is transferred.

- Form 1

  Internal network analyzer format, 6 bytes per data point. The array is preceded by a four byte header. The first two bytes represent the string "#A", which is the standard block header. The second two bytes are an integer holding the number of bytes in the block to follow. This means that a 201 point transfer is 1210 bytes. Form 1 is intended for rapid data transfers to and from the computer, not for manipulation or subsequent processing by the computer.

- Form 2

  IEEE 32-bit floating point format, 8 bytes per data point. The array is preceded by a four byte header, The first two bytes represent the string "#A", which is the standard block header. The second two bytes are an integer holding the number of bytes in the block to follow. Each number consists of a sign bit, 8 bit signed exponent, and 23 bit mantissa. Two numbers make up a single data point. A 201 point transfer is 1612 bytes.

- Form 3

  IEEE 64-bit floating point format, 16 bytes per data point. The array is preceded by a four byte header, The first two hytes represent the string "#A", which is the standard block header. The second two bytes are an integer holding the numher of bytes in the block to follow. Each number consists of a sign bit, 11 bit signed exponent, and 52 bit mantissa. Two numhers make up a single data point. A 201 point transfer is 2220 hytes.

- Form 4

  ASCII data transfer format. In this mode, each number is sent as a 24 character string, each character being a digit, sign, or decimal point. Since there are two numbers per point, a 201-point transfer is 9648 bytes.

- Form 5

  MS-DOS® personal computer format. This mode is a modification of IEEE 32-bit floating point format with the byte order reversed. The array is preceded by a four byte header. The first two bytes represent the string "#A", the standard block header. The second

two bytes are an integer holding the number of bytes in the block to follow, with the least significant byte preceding the most significant byte. Like Form 3, there are 8 bytes per data point, but the least significant byte precedes the more significant. Thus, in this format, an MS-DOS® PC can store data internally without reformatting it. A 201 point transfer is 1612 bytes.

Data Levels

Different levels of data can be read out of the network analyzer (Refer to Figure 1-3).

- Raw data

  The basic measurement data, which depends on the selection of the stimulus parameters, IF averaging, and IF bandwidth. If a full 2-port measurement calibration is ON, there are four raw arrays kept: one for each raw S-parameter. The data is read out with the commands `OUTPRAW{1-4}`. Otherwise, only raw array 1 is available, and it holds the current parameter. If a 2-port calibration is ON the four arrays correspond to S_{11}, S_{21}, S_{12}, and S_{22} respectively. This data is in real/imaginary pairs.

- Error corrected data

  This is the raw data with error correction applied. The array is for the currently measured parameter, and is in real/imaginary pairs. The error corrected data is read out with `OUTPDATA`. `OUTPMEMO` reads the trace memory if available, which is also error corrected. Neither raw nor error corrected data include the effects of post-processing functions, such as electrical delay offset or trace math.

- Formatted data

  This is the array of data being displayed. It includes the effects of all post-processing functions such as electrical delay, and the units of the array read out depends on the current display format. Refer to Table 2-1 for various units as a function of display format. The formatted data is read out with `OUTPFORM`.

- Calibration coefficients

  The results of a calibration are arrays of calibration coefficients (also called error coefficients) which are used in the error correction routines. Each array corresponds to a specific error term in the error model. The calibration coefficients are read out with `OUTPCALC{01|12}`.

Formatted data is generally the most useful, being the same information seen on the display. However, if the post-processing results are not necessary, error corrected data may be more desirable. Error corrected data also gives you the opportunity to load the data into the instrument and apply post-processing at a later time.

Data Transfer Using ASCII Transfer Format (Form 4)

When Form 4 is used, each number is sent as a 24 character string, each character being a digit, or decimal point. Since there are two numbers per point, a 201-point transfer in Form 4 takes 9648 bytes.

```
10    !
20    ! Data Transfer using ASCII Transfer Format
30    !
40    OPTION BASE 1
50    Hp8720=716
60    ABORT 7
70    CLEAR Hp8720
80    !
90    OUTPUT Hp8720;"PRES;"
100   OUTPUT Hp8720;"CHAN1; S21; LOGM;"
110   INPUT "Enter center frequency (Hz)",F_cent
120   INPUT "Enter frequency span (Hz)",F_span
130   OUTPUT Hp8720;"CENT ";F_cent
140   OUTPUT Hp8720;"SPAN ";F_span
150   !
160   OUTPUT Hp8720;"OPC?;"
170   OUTPUT Hp8720;"SING;"
180   ENTER Hp8720;Stat
190   !
200   OUTPUT Hp8720;"POIN?;"
210   ENTER Hp8720;Nump
220   ALLOCATE Dat(Nump,2),Stim(Nump)
230   OUTPUT Hp8720;"FORM4;"
240   !
250   OUTPUT Hp8720;"OUTPFORM;"
260   ENTER Hp8720;Dat(*)
270   !
280   F_start=F_cent-F_span/2
290   F_incre=F_span/(Nump-1)
300   !
310   FOR I=1 TO Nump
320     Stim(I)=F_start+F_incre*(I-1)
330     PRINT Stim(I);"Hz",Dat(I,1);"dB"
340   NEXT I
350   DEALLOCATE Dat(*),Stim(*)
360   END
```

Figure 2-6. Sample Program: Data Transfer using ASCII Transfer Format (Form 4)

Program explanation

Line 40	Specifies the default lower bound of arrays to 1.
Lines 200 and 210	Finds out how many points to expect.
Line 220	Create arrays to hold the trace data and the stimulus data.
Line 230	Tells the network analyzer to use the ASCII transfer format.
Line 250	Requests the formatted trace data.
Line 260	Transfers the data from the network analyzer to the computer, and puts it in the receiving array.
Lines 310 through 340	Calculates the stimulus value and prints the data. Also, it is possible to read the frequencies directly out of the network analyzer using the OUPTLIML command, which reports the limit test results by transmitting the stimulus point tested, a number indicating the limit test results, and then the upper and lower limits at the stimulus point. The number inidcating the limit results is a -1 for no test, 0 for fail and 1 for pass. If there are no upper or lower limits set, the output for the limits is zeroes. To try this, delete line 320 and edit lines 220, 280, 290, and 330 as follows: 220 ALLOCATE Dat(Nump,2),Stim(Nump,4) 280 OUTPUT Hp8720;"OUTPLIML;" 290 ENTER Hp8720;Stim(*) 330 PRINT Stim(I,1);"Hz",Dat(I,1);"dB"
Line 350	Deallocates memory space.

Data Transfer using IEEE 64-bit Floating Point Format (Form 3)

Form 3 utilizes the IEEE 64-bit Floating Point Format for real numbers. The data transfer begins with a four byte header, the first two bytes correspond to the ASCII characters "#A" indicating a fixed length block transfer. The second two byte pair form an integer containing the number of bytes in the block to follow. Since Form 3 requires only 8 bytes for each number (compared to 24 bytes for ASCII Form 4), the data is transferred faster.

The transfer of data when using Form 3 is further enhanced by defining an I/O path with formatting OFF. Note the use of the ASSIGN statement below.

```
10    !
20    ! Data Transfer using IEEE 64-bit Floating Point Format
30    !
40    OPTION BASE 1
50    Hp8720=716
60    ABORT 7
70    CLEAR Hp8720
80    !
90    OUTPUT Hp8720;"PRES;"
100   OUTPUT Hp8720;"CHAN1; S21; LOGM;"
110   INPUT "Enter center frequency (Hz)",F_cent
120   INPUT "Enter frequency span (Hz)",F_span
130   OUTPUT Hp8720;"CENT ";F_cent
140   OUTPUT Hp8720;"SPAN ";F_span
150   !
160   OUTPUT Hp8720;"OPC?;"
170   OUTPUT Hp8720;"SING;"
180   ENTER Hp8720;Stat
190   !
200   OUTPUT Hp8720;"POIN?;"
210   ENTER Hp8720;Nump
220   ALLOCATE Dat(Nump,2),Stim(Nump)
230   INTEGER Hdr,Lgth
240   ASSIGN @Data TO Hp8720;FORMAT OFF
250   OUTPUT Hp8720;"FORM3;"
260   !
270   OUTPUT Hp8720;"OUTPFORM;"
280   ENTER @Data;Hdr,Lgth,Dat(*)
290   !
300   F_start=F_cent-F_span/2
310   F_incre=F_span/(Nump-1)
320   !
330   FOR I=1 TO Nump
340     Stim(I)=F_start+F_incre*(I-1)
350     PRINT Stim(I);"Hz",Dat(I,1);"dB"
360   NEXT I
370   DEALLOCATE Dat(*),Stim(*)
380   ASSIGN @Data TO *
```

```
390    END
```

Figure 2-7. Sample Program: Data Transfer using IEEE 64-bit Floating Point Format (Form 3)

Program explanation

Line 240	Sets up the I/O path; "FORMAT OFF" matches up the computer's real number format (IEEE 64-bit) to Form 3.
Line 250	Tells network analyzer to output data using Form 3.
Line 280	Enters the header, followed by the data.
Line 380	Closes the I/O path.

Application Example

The following example is to measure the transmission parameter a bandpass filter and to get the typical parameters: −3 dB bandwidth, Center frequency, and Insertion loss.

```
10    !
20    ! Bandpass Filter Test
30    !
40    Hp8720=716
50    ABORT 7
60    CLEAR Hp8720
70    !
80    OUTPUT Hp8720;"PRES;"
90    OUTPUT Hp8720;"CHAN1; S21; LOGM;"
100   INPUT "Enter center frequency (Hz)",F_cent
110   INPUT "Enter frequency span (Hz)",F_span
120   OUTPUT Hp8720;"CENT ";F_cent
130   OUTPUT Hp8720;"SPAN ";F_span
140   !
150   OUTPUT Hp8720;"HOLD;"
160   OUTPUT Hp8720;"CALK35MM;"
170   OUTPUT Hp8720;"CALIRESP;"
180   INPUT "Connect THRU, then press [Return].",Dum$
190   OUTPUT Hp8720;"OPC;"
200   OUTPUT Hp8720;"STANC;"
210   GOSUB Op_end
220   OUTPUT Hp8720;"RESPDONE;"
230   INPUT "Cal completed. Connect DUT, then press [Return].",Dum$
240   !
250   OUTPUT Hp8720;"OPC;"
260   OUTPUT Hp8720;"SING;"
270   GOSUB Op_end
280   !
290   OUTPUT Hp8720;"MARK1;"
300   OUTPUT Hp8720;"SEAMAX;"
310   OUTPUT Hp8720;"OUTPMARK;"
320   ENTER Hp8720;Loss,Val2,Stim
330   !
340   OUTPUT Hp8720;"DELR1;"
350   OUTPUT Hp8720;"WIDV -3;"
360   OUTPUT Hp8720;"WIDTON;"
370   OUTPUT Hp8720;"OUTPMWID;"
380   ENTER Hp8720;Bw,Cent,Q
390   !
400   PRINT "-3 dB bandwidth:",Bw;"Hz"
410   PRINT "Center frequency:",Cent;"Hz"
420   PRINT "Insertion loss:",Loss;"dB"
430   STOP
440   !
450 Op_end:  !
```

```
460   REPEAT
470     OUTPUT Hp8720;"ESB?"
480     ENTER Hp8720;Stat
490   UNTIL BIT(Stat,0)
500   RETURN
510   END
```

Figure 2-8. Sample Program: Application Example (Bandpass Filter Test)

Program explanation

Lines 80 through 130	Sets up measurement.
Lines 150 through 230	Does response calibration.
Lines 250 through 270	Takes one sweep of data.
Lines 290 through 320	Takes the insertion loss value using the marker search function.
Lines 340 through 380	Takes the −3 dB bandwidth value and the center frequency value using the bandwidth search function.

3

Advanced Programming Examples

Using List Frequency Mode

The list frequency mode lets you select the specific points or frequency spacing between points at which measurements are to be made. Sampling specific points reduces the measurement time since additional time is not spent measuring device characteristics at unnecessary frequencies.

This example shows how to create a list frequency table and transmit it to the network analyzer. The command sequence for entering a list frequency table imitates the key sequence followed when entering a table from the front panel: there is a command for every key press. Editing a segment is also the same as the key sequence, but the network analyzer automatically reorders each edited segment in order of increasing start frequency.

This example takes advantage of the computer's capabilities to simplify creating and editing the table. The table is entered and completely edited before being transmitted to the network analyzer. To simplify the programming task, options such as entering step size are not included.

A single segment of the list can be displayed and measured using the SSEGn command, where n=the segment number. The all segments command ASEG returns the network analyzer to the full frequency list.

```
10    !
20    ! Using List Frequency Mode
30    !
40    OPTION BASE 1
50    Hp8720=716
60    ABORT 7
70    CLEAR Hp8720
80    !
90    INPUT "Enter number of segments",Numb
100   ALLOCATE Table(Numb,3)
110   !
120   PRINTER IS 1
130   OUTPUT 2;CHR$(255)&"K";
140   PRINT USING "10A,10A,10A,20A";"Segment",
       "Start(Hz)","Stop(Hz)", "Number of Points"
150   !
160   FOR I=1 TO Numb
170     GOSUB Loadpoin
180   NEXT I
190   !
200   LOOP
210     INPUT "Do you want to edit?  (Y/N)",An$
```

```
220   EXIT IF An$="N" OR An$="n"
230     INPUT "Enter segment number",I
240     GOSUB Loadpoin
250   END LOOP
260   !
270   OUTPUT Hp8720;"PRES;"
280   OUTPUT Hp8720;"CHAN1; S21; LOGM;"
290   !
300   OUTPUT Hp8720;"EDITLIST;"
310   OUTPUT Hp8720;"CLEL;"
320   FOR I=1 TO Numb
330     OUTPUT Hp8720;"SADD;"
340     OUTPUT Hp8720;"STAR ";Table(I,1)
350     OUTPUT Hp8720;"STOP ";Table(I,2)
360     OUTPUT Hp8720;"POIN ";Table(I,3)
370     OUTPUT Hp8720;"SDON;"
380   NEXT I
390   OUTPUT Hp8720;"EDITDONE;"
400   OUTPUT Hp8720;"LISFREQ;"
420   !
430   OUTPUT Hp8720;"OPC;"
440   OUTPUT Hp8720;"SING;"
450   REPEAT
460     OUTPUT Hp8720;"ESB?"
470     ENTER Hp8720;Stat
480   UNTIL BIT(Stat,0)
490   OUTPUT Hp8720;"AUTO;"
500   STOP
510   !
520 Loadpoin:   !
530   INPUT "Enter start frequency (Hz)",Table(I,1)
540   INPUT "Enter stop frequency (Hz)",Table(I,2)
550   INPUT "Enter number of points",Table(I,3)
560   IF Table(I,3)=1 THEN Table(I,2)=Table(I,1)
570   PRINT TABXY(0,I+1);I;TAB(10);Table(I,1);TAB(20);
       Table(I,2);TAB(35);Table(I,3)
580   RETURN
590   END
```

Figure 3-1. Sample Program: Using List Frequency Mode

Program explanation

Line 90	Finds out how many segments to expect.
Line 100	Creates a table to hold the segments. Keeps start frequency, stop frequency, and number of points.
Lines 120 through 140	Clears the screen and print the table header.
Lines 160 through 180	Prompts for the start, stop, and number of points for each segment.
Lines 200 through 250	Edits the table until editing is no longer needed.
Line 300	Activates the frequency list edit mode, and opens the list frequency table for editing.
Line 310	Deletes any existing segments.
Lines 320 through 380	Enters the segment values.
Line 390	Closes the table.
Line 400	Turns on list frequency mode.
Line 410	Displays the trace for only the listed frequency ranges.
Lines 520 through 580	Enters in a segment.
Lines 530 through 550	Enters the segment values.
Line 560	Makes the stop frequency equal to the start frequency to avoid ambiguity, if only one point is in the segment.
Line 570	Prints the segment out.

Using Limit Lines to Perform Limit Testing

This example shows how to create a limit table and transmit it to the network analyzer. The command sequence for entering a limit table imitates the key sequence followed when entering a table from the front panel: there is a command for every key press. Editing a limit is also the same as the key sequence, but remember that the network analyzer automatically reorders the table in order of increasing start frequency.

This example takes advantage of the computer's capabilities to simplify creating and editing the table. The table is entered and completely edited before being transmitted to the network analyzer. To simplify the programming task, options such as entering offsets are not included.

```
10    !
20    ! Setting up Limit Lines
30    !
40    OPTION BASE 1
50    Hp8720=716
60    ABORT 7
70    CLEAR Hp8720
80    !
90    OUTPUT Hp8720;"PRES;"
100   OUTPUT Hp8720;"CHAN1; S21; LOGM;"
110   INPUT "Enter start frequency (Hz)",F_start
120   INPUT "Enter stop frequency (Hz)",F_stop
130   OUTPUT Hp8720;"STAR ";F_start
140   OUTPUT Hp8720;"STOP ";F_stop
150   !
160   INPUT "Enter number of limits",Numb
170   ALLOCATE Table(Numb,3)
180   !
190   PRINTER IS 1
200   OUTPUT 2;CHR$(255)&"K";
210   PRINT USING "10A,15A,15A,15A";"Segment",
       "Stimulus(Hz)","Upper(dB)","Lower(dB)"
220   !
230   FOR I=1 TO Numb
240     GOSUB Loadlimit
250   NEXT I
260   !
270   LOOP
280     INPUT "Do you want to edit? (Y/N)",An$
290   EXIT IF An$="N" OR An$="n"
300     INPUT "Enter segment number",I
310     GOSUB Loadlimit
320   END LOOP
330   !
340   OUTPUT Hp8720;"EDITLIML;"
350   OUTPUT Hp8720;"CLEL;"
360   FOR I=1 TO Numb
370     OUTPUT Hp8720;"SADD;"
380     OUTPUT Hp8720;"LIMS ";Table(I,1)
```

```
390      OUTPUT Hp8720;"LIMU ";Table(I,2)
400      OUTPUT Hp8720;"LIML ";Table(I,3)
410      OUTPUT Hp8720;"SDON;"
420    NEXT I
430    !
440    OUTPUT Hp8720;"EDITDONE"
450    OUTPUT Hp8720;"LIMILINEON"
460    OUTPUT Hp8720;"LIMITESTON"
470    DEALLOCATE Table(*)
480    STOP
490    !
500 Loadlimit:  !
510    INPUT "Enter stimulus value (Hz)",Table(I,1)
520    INPUT "Enter upper limit value (dB)",Table(I,2)
530    INPUT "Enter lower limit value (dB)",Table(I,3)
540    PRINT TABXY(0,I+1);I;TAB(11);Table(I,1);TAB(27);
        Table(I,2);TAB(42);Table(I,3)
550    RETURN
560    END
```

Figure 3-2. Sample Program: Setting up Limit Lines

Program explanation

Line 160	Finds out how many limits to expect.
Line 170	Creates a table to hold the limits. It will contain the stimulus value (frequency), the upper limit value, and the lower limit value.
Lines 190 through 210	Clears the screen and prints the table header.
Lines 230 through 250	Reads in each segment.
Lines 270 through 320	Edits the table until editing is no longer needed.
Line 340	Begins editing the limit line table.
Line 350	Deletes any existing limits.
Lines 360 through 420	Enters the segment values.
Line 440	Closes the table.
Line 450	Displays the limits.
Line 460	Activates the limit testing.
Lines 500 through 550	Enters a segment.

Storing and Recalling Instrument Status

Coordinating disk storage

This example shows how to save and recall the instrument status in the disk installed in the built-in disk drive.

```
10   !
20   ! Storing Instrument States
30   !
40   DIM Err$[50]
50   Hp8720=716
60   ABORT 7
70   CLEAR Hp8720
80   !
90   OUTPUT Hp8720;"PRES;"
100  OUTPUT Hp8720;"CHAN1; S21; LOGM;"
110  INPUT "Enter center frequency (Hz)",F_cent
120  INPUT "Enter frequency span (Hz)",F_span
130  OUTPUT Hp8720;"CENT ";F_cent
140  OUTPUT Hp8720;"SPAN ";F_span
150  !
160  INPUT "File name? (up to 8 char.)",Name$
170  OUTPUT Hp8720;"USEPASC;"
180  OUTPUT Hp8720;"TITF1""";Name$;""";STOR1;"
190  PASS CONTROL Hp8720
200  !
210  STATUS 7,6;Stat
220  IF NOT BIT(Stat,6) THEN GOTO 210
230  !
240  INPUT "Save done. Press [Return] to recall.",Dum$
250  !
260  OUTPUT Hp8720;"PRES;"
270  OUTPUT Hp8720;"USEPASC;"
280  OUTPUT Hp8720;"TITF1""";Name$;""";LOAD1;"
290  PASS CONTROL Hp8720
300  STATUS 7,6;Stat
310  IF NOT BIT(Stat,6) THEN GOTO 300
320  !
330  DISP "Done."
340  END
```

Figure 3-3. Sample Program: Storing Instrument States

Program explanation

Line 160	Gets the name of the file to create.
Line 170	Enable the network analyzer to use the pass control capability.
Line 180	Saves the instrument states and the calibration coefficients with the file name. The file name must be preceded and followed by double quotation marks, and the only way to do that with an OUTPUT statement is to use two sets of quotation marks: "".
Lines 190	The computer passes Active Controller to the network analyzer.
Lines 210 and 220	Wait for Active Controller status to return to the computer.
Line 260 through 310	Preset the network analyzer and recall the previously stored file.

Reading Calibration Data

This example demonstrates how to read measurement calibration data out of the network analyzer, and how to put it back into the network analyzer.

The data used to perform measurement error correction is stored inside the network analyzer in up to twelve calibration coefficient arrays. Each array is a specific error coefficient, and is stored and transmitted as an error corrected data array: each point is a real/imaginary pair, and the number of points in the array is the same as the number of points in the sweep. The four data formats also apply to the transfer of calibration coefficient arrays. *HP-IB Programming Reference* specifies where the calibration coefficients are stored for different calibration types.

The computer can read out the error coefficients using the `OUTPCALC{01-12}` commands. Each calibration type uses only as many arrays as needed, starting with array 1. Therefore, it is necessary to know the type of calibration about to be read out: attempting to read an array not being used in the current calibration causes the "REQUESTED DATA NOT CURRENTLY AVAILABLE" error message.

The computer can also input calibration coefficients to the network analyzer. To do this, declare the type of calibration data about to stored in the network analyzer just as if you were about to perform that calibration. Then, instead of calling up different classes, transfer the calibration coefficients using the `INPUCALC{01-12}` commands. When all the coefficients are in the network analyzer, activate the calibration by issuing the mnemonic `SAVC`, and have the network analyzer take a sweep.

This example reads the response calibration coefficients into a very large array, from which they can be examined, modified, stored, or put back into the network analyzer.

```
10    !
20    !  Reading calibration data.
30    !  It demonstrates how to read calibration data out
40    !  of the network analyzer, and how to put it back in.
50    !
60    !  The program will handle any type of calibration,
70    !  and any number of points.
80    !
90    !
100   Hp8720=716
110   ABORT 7
120   CLEAR Hp8720
130   DATA "CALIRESP",1,"CALIRAI",2,"CALIS111",3
140   DATA "CALIS221",3,"CALIFUL2",12,"CALITRL2",12
150   DATA "NOOP",0
160   INTEGER Hdr,Lgth,I,J
170   ASSIGN @Data TO Hp8720;FORMAT OFF
180   !
190   READ Calt$,Numb
200   IF Numb=0 THEN GOTO 510
210   OUTPUT Hp8720;Calt$;"?;"
220   ENTER Hp8720;Active
230   IF NOT Active THEN GOTO 190
```

```
240   !
250   DISP Calt$,Numb
260   OUTPUT Hp8720;"FORM3;POIN?;"
270   ENTER Hp8720;Poin
280   ALLOCATE Cal(1:Numb,1:Poin,1:2)
290   FOR I=1 TO Numb
300     OUTPUT Hp8720 USING "K,ZZ";"OUTPCALC",I
310     ENTER @Data;Hdr,Lgth
320     FOR J=1 TO Poin
330       ENTER @Data;Cal(I,J,1),Cal(I,J,2)
340     NEXT J
350   NEXT I
360   !
370   OUTPUT Hp8720;"CORROFF;"
380   !
390   INPUT "Press [Return] to retransmit calibration data.",Dum$
400   OUTPUT Hp8720;Calt$,";"
410   FOR I=1 TO Numb
420     DISP "TRANSMITTING ARRAY: ",I
430     OUTPUT Hp8720 USING "K,ZZ";"FORM3;INPUCALC",I
440     OUTPUT @Data;Hdr,Lgth
450     FOR J=1 TO Poin
460       OUTPUT @Data;Cal(I,J,1),Cal(I,J,2)
470     NEXT J
480   NEXT I
490   OUTPUT Hp8720;"SAVC;"
500   OUTPUT Hp8720;"CONT;"
510   DISP "DONE"
520   END
```

Figure 3-4. Sample Program: Reading calibration data

Program explanation

Line 130 through 150	Set up the data base of possible calibrations, and the number of arrays associated with each calibration.
Line 190 through 230	Get a calibration type and the corresponding number of arrays for that calibration type. If correction was not one, exit the program. Query the network analyzer to determine if the calibration type is active; if not, loop back to read another type.
Line 250	Display the active calibration type and number of arrays.
Line 260 through 280	Establish Form 3 as the data transfer format; query the number of points. Allocate the required memory space based on the number of points.
Line 290 through 350	Request output of the appropriate calibration array; get the file header and calibration data for each array.
Line 400	Set up the calibration type for the arrays about to be loaded.

Line 410 through 480	Load each calibration array into the network analyzer.
Line 490	End of loading calibration array(s), save in internal network analyzer memory, and turn correction on.
Line 500	Set sweep to continuous to show that calibration arrays have been properly loaded.

4

Miscellaneous Programming Examples

Controlling Peripherals

The purpose of this section is to demonstrate how to coordinate printers or plotters with the network analyzer.

The network analyzer has three operating modes with respect to HP-IB, as set under the LOCAL menu: System controller, Talker/Listener, and Use Pass Control. The System Controller mode is used when no controller is present, and controls peripherals directly. The Talker/Listener mode is the most common most in which the computer controls the network analyzer. In the Use Pass Control mode, the network analyzer acts like a Talker/Listener, but additionally can take active control from the computer (System Controller) so that the network analyzer can directly control peripherals.

Note that the network analyzer assumes that the address of the computer is correctly stored in its HP-IB addresses menu under the ADDRESS: CONTROLLER entry. If this address is incorrect, control will not return to the computer.

This example shows control of a plotter with Talker/Listener mode.

```
10    !
20    !  Operation using Talker/listener mode.
30    !
40    Hp8720=716
50    OUTPUT Hp8720;"OUTPPLOT;"
60    SEND 7;UNL LISTEN 5 TALK 16 DATA
70    DISP "PLOTTING"
80    STATUS 7,7;Stat
90    IF NOT BIT(Stat,11) THEN GOTO 80
100   DISP "DONE"
110   END
```

Figure 4-1. Sample Program: Using Talker/Listener Mode

Program explanation

Line 50 — Command the network analyzer to plot.

Line 60 — Use the HP Basic commands for HP-IB control to establish a data path from the network analyzer to the plotter. SEND 7 means that bus control commands will be sent from the Active Controller with select code 7. UNL forces all talker/listener instruments to "unlisten." LISTEN 5 commands the device at address 5 (the plotter) to now "listen" (wait for data). TALK 16 commands the network analyzer

to "talk" (send the data). DATA forces the HP-IB from "command" mode to "data" mode.

Line 70 — Display that plotting is taking place, and provide momentary delay to prevent interrogation of status register 7.

Lines 80 through 90 — Wait for the network analyzer to assert EOI, indicating end of transmission of data.

Using Pass Control Mode

This example shows control of a printer with Use Pass Control mode.

```
10    !
20    !  Operation using pass control mode.
30    !
40    Hp8720=716
50    OUTPUT Hp8720;"CLES;ESE2;"
60    OUTPUT Hp8720;"USEPASC;PRINALL;"
70    Stat=SPOLL(Hp8720)
80    IF NOT BIT(Stat,5) THEN GOTO 70
90    PASS CONTROL Hp8720
100   DISP "PRINTING"
110   STATUS 7,6;Hpib
120   IF NOT BIT(Hpib,6) THEN GOTO 110
130   DISP "DONE"
140   END
```

Figure 4-2. Sample Program: Using Pass Control Mode

Program explanation

Line 50 — Clear the status reporting system. Enable the Request Active Control bit in the event status register.

Line 60 — Enable Use Pass Control. Request print.

Lines 70 through 80 — Wait until the network analyzer requests control.

Line 90 — Passes active control to the network analyzer.

Line 110 through 120 — Waits until the print is finished and the control is returned.

Status and Error Reporting

The network analyzer has a status reporting mechanism that gives information about specific functions and events inside the network analyzer. The status byte is an 8-bit register with each bit summarizing the state of one aspect of the network analyzer. For example, the error queue summary bit will always be set if there are any errors in the queue. The value of the status byte can be read with the `SPOLL` statement. This command does not automatically put the network analyzer into the remote mode, thus giving the operator access to the network analyzer front panel functions. The status byte can also be read using the `OUTPSTAT` command, but the network analyzer will be put into remote mode. Reading the status byte does not affect its value.

The status byte also summarizes two event status registers that monitor specific conditions inside the network analyzer. The status byte also has a bit that is set when the network analyzer is issuing a service request over HP-IB, and a bit that is set when the network analyzer has data to send out over HP-IB. Refer to the *HP-IB Programming Reference* for a definition of the status registers.

The error queue holds up to 20 instrument errors and warnings in the order that they occurred. Each time the network analyzer detects an error condition and displays a warning message on the CRT, it also puts the error in the error queue. If there are any errors in the queue, bit 3 of the status byte will be set. The errors can be read from the queue with the `OUTPERRO` command, which causes the network analyzer to transmit the error number and the error message of the oldest error in the queue.

It is also possible to generate interrupts using the status reporting mechanism. The status byte bits can be enabled to generate a service request (SRQ) when set. The computer can in turn be set up to generate an interrupt on the SRQ.

To be able to generate an SRQ, a bit in the status byte has to be enabled using `SRE` *n* (Status Register Enable Mask). A one in a bit position enables that bit in the status byte. Therefore, `SRE 8` enables an SRQ on bit 3, check error queue, since 8 equals 00001000 in binary representation. That means that whenever an error is put into the error queue and bit 3 gets set, and the SRQ line is asserted. The only way to clear the SRQ is to disable bit 3, re-enable bit 3, or read out all the errors from the queue.

A bit in the event status register can be enabled so that it is summarized by bit 5 of the status byte. If any bit is enabled in the event status register, bit 5 of the status byte will also be set. For example, `ESE 66` (Event Status Enable Mask) enables bits 1 and 6 of the event status register, since 66 equals 01000010 in binary representation. Therefore, whenever active control is requested or a front panel key is pressed, bit 5 of the status byte will be set. Similarly, `ESNB` *n* enables bits in event status register B so that they will be summarized by bit 2 in the status byte.

To generate an SRQ from an event status register, enable the desired event status register bit. Then enable the status byte to generate an SRQ. For instance, `*ESE 32` and `*SRE 32` enable the syntax error bit, so that when the syntax error bit is set, the summary bit in the status byte will be set, and it enables an SRQ on bit 5 of the status byte.

```
10    !
20    ! Generating Interrupts
30    !
40    Hp8720=716
```

```
45    DIM Err$[50]
50    !
60    OUTPUT Hp8720;"CLES;"
70    OUTPUT Hp8720;"ESE 32;"
80    OUTPUT Hp8720;"SRE 32;"
90    !
100   ON INTR 7 GOSUB Err_report
110   ENABLE INTR 7;2
120   !
130   LOOP
140   END LOOP
150   STOP
160   !
170 Err_report:!
180   Stat=SPOLL(Hp8720)
190   OUTPUT Hp8720;"ESR?"
200   ENTER Hp8720;Estat
210   PRINT "Syntax error detected."
220   !
230   OUTPUT Hp8720;"OUTPERRO;"
240   ENTER Hp8720;Err,Err$
250   PRINT Err,Err$
260   IF Err0 THEN GOTO 230
270   ENABLE INTR 7
280   RETURN
290   END
```

Figure 4-3. Sample Program: Generating Interrupts

Program explanation

Line 60	Clears the status reporting system.
Line 70	Enables bit 5 of the event status register.
Line 80	Enables bit 5 of the status byte so that an SRQ will generated on a syntax error.
Line 100	Tells the computer where to branch it gets the interrupt.
Line 110	Tells the computer to enable an interrupt from interface 7 (HP-IB) when value 2 (bit 1: SRQ bit) of the interrupt register is set. A branch to `Err_report` will disable the interrupt, so the return from `Err_report` re-enables it. Since there may be more than one instrument on the bus capable of generating an SRQ, it will be necessary to serial poll to determine that the network analyzer issued the SRQ. A branch to `Err_report` will disable the interrupt, so the return from `Err_report` re-enable it.
Line 130 and 140	Loop until interrupted.

Line 180	Clears the SRQ bit of the status byte. At the point, the variable `stat` could have been checked to see if, in fact, the network analyzer requested service.
Lines 190 and 200	Reads the register to clear the bit.
Lines 230 through 260	Instructs the network analyzer to output the error number and the error message, and print them. The output will loop until no errors (Err is 0).

Modifying Calibration Kit

The network analyzer has several calibration kit definitions built into the firmware. These can be called by using the CALKnnnn command. For example, the Type N 50 ohm kit (HP 85054D) can be selected using CALKN50.

For other calibration kits, and customizing the default kits, the following program shows how to perform the definition modification automatically.

```
10    ! Creating an X band Calibration Kit Definition
20    ! for the HP X11644A
30    !
90    ASSIGN @Ana TO 716
100   Minf=6.555E+9                               ! MIN. FREQUENCY
110   Maxf=1.3111E+10                             ! MAX. FREQUENCY
120   OUTPUT @Ana;"PRES;VELOFACT0.99968;"
130   OUTPUT @Ana;"SETZ1;";                       ! Set system imped to 1 ohm
140   ! Define standard #1
150   OUTPUT @Ana;"MODI1;";                       ! Modify cal kit #1 (8720)
160   OUTPUT @Ana;"DEFS1;";                       ! Begin defining std # 1
170   OUTPUT @Ana;"STDTSHOR;";                    ! std #1 will be a short
180   OUTPUT @Ana;"OFSD0;";                       ! offset delay = 0 ps
190   OUTPUT @Ana;"OFSL0;";                       ! offset loss = 0
200   OUTPUT @Ana;"OFSZ1;";                       ! offset imped = 1 ohm
210   OUTPUT @Ana;"MINF",Minf;"HZ";
220   OUTPUT @Ana;"MAXF",Maxf;"HZ";
230   OUTPUT @Ana;"WAVE;";                        ! waveguide standard
240   OUTPUT @Ana;"STDD;";                        ! standard defined
250   OUTPUT @Ana;"LABS""SHORT"";";               ! label standard
260   ! Define standard #2
270   OUTPUT @Ana;"DEFS2;";                       ! Begin defining std # 2
280   OUTPUT @Ana;"STDTSHOR;";                    ! std #2 will be a short
290   OUTPUT @Ana;"OFSD32.633PS;";                ! offset delay
300   OUTPUT @Ana;"OFSL0;";                       ! offset loss = 0
310   OUTPUT @Ana;"OFSZ1;";                       ! offset imped = 1 ohm
320   OUTPUT @Ana;"MINF",Minf;"HZ";
330   OUTPUT @Ana;"MAXF",Maxf;"HZ";
340   OUTPUT @Ana;"WAVE;";                        ! waveguide standard
350   OUTPUT @Ana;"STDD;";                        ! standard defined
360   OUTPUT @Ana;"LABS""1/4 OFFSET"";";          ! label standard
370   ! Define standard #3
380   OUTPUT @Ana;"DEFS3;";                       ! Begin defining std # 3
390   OUTPUT @Ana;"STDTLOAD;";                    ! std #3 will be a load
400   OUTPUT @Ana;"FIXE;"                         ! fixed load
410   OUTPUT @Ana;"OFSD0;";                       ! offset delay = 0
420   OUTPUT @Ana;"OFSL0;";                       ! offset loss = 0
430   OUTPUT @Ana;"OFSZ1;";                       ! offset imped = 1 ohm
440   OUTPUT @Ana;"MINF",Minf;"HZ";
450   OUTPUT @Ana;"MAXF",Maxf;"HZ";
460   OUTPUT @Ana;"WAVE;";                        ! waveguide standard
470   OUTPUT @Ana;"STDD;";                        ! standard defined
```

```
480  OUTPUT @Ana;"LABS""FIXED"";";              ! label standard
490  ! Define standard #4
500  OUTPUT @Ana;"DEFS4;";                      ! Begin defining std # 4
510  OUTPUT @Ana;"STDTDELA;";                   ! std #4 will be a THRU
520  OUTPUT @Ana;"OFSD0;";                      ! offset delay = 0
530  OUTPUT @Ana;"OFSL0;";                      ! offset loss = 0
540  OUTPUT @Ana;"OFSZ1;";                      ! offset imped = 1 ohm
550  OUTPUT @Ana;"MINF",Minf;"HZ";
560  OUTPUT @Ana;"MAXF",Maxf;"HZ";
570  OUTPUT @Ana;"WAVE;";                       ! waveguide standard
580  OUTPUT @Ana;"STDD;";                       ! standard defined
590  OUTPUT @Ana;"LABS""THRU"";";               ! label standard
600  ! Define standard #5
610  OUTPUT @Ana;"DEFS5;";                      ! Begin defining std # 5
620  OUTPUT @Ana;"STDTDELA;";                   ! std #5 will be a THRU
630  OUTPUT @Ana;"OFSD32.633PS;";               ! offset delay
640  OUTPUT @Ana;"OFSL0;";                      ! offset loss = 0
650  OUTPUT @Ana;"OFSZ1;";                      ! offset imped = 1 ohm
660  OUTPUT @Ana;"MINF",Minf;"HZ";
670  OUTPUT @Ana;"MAXF",Maxf;"HZ";
680  OUTPUT @Ana;"WAVE;";                       ! waveguide standard
690  OUTPUT @Ana;"STDD;";                       ! standard defined
700  OUTPUT @Ana;"LABS""1/4 DELAY"";";          ! label standard
710  ! Specify the standards for a given class
720  OUTPUT @Ana;"SPECRESP1,2,4;";              ! response cal
730  OUTPUT @Ana;"SPECRESI1,2,4;";              ! response & isol
740  OUTPUT @Ana;"SPECS11A1;";                  ! s11 class "A"
750  OUTPUT @Ana;"SPECS11B2;";                  ! s11 class "B"
760  OUTPUT @Ana;"SPECS11C3;";                  ! s11 class "C"
770  OUTPUT @Ana;"SPECS22A1;";                  ! s22 class "A"
780  OUTPUT @Ana;"SPECS22B2;";                  ! s22 class "B"
790  OUTPUT @Ana;"SPECS22C3;";                  ! s22 class "C"
800  OUTPUT @Ana;"SPECFWDT4;";                  ! forward trans.
810  OUTPUT @Ana;"SPECFWDM4;";                  ! forward match
820  OUTPUT @Ana;"SPECREVT4;";                  ! reverse trans.
830  OUTPUT @Ana;"SPECREVM4;";                  ! reverse match
840  OUTPUT @Ana;"CLAD;";                       ! class definitions done
850  ! Label specific classes
860  OUTPUT @Ana;"LABES11A""SHORT"";";          ! s11 class "A"
870  OUTPUT @Ana;"LABES11B""1/4 SHORT"";";      ! s11 class "B"
880  OUTPUT @Ana;"LABES11C""FIXED LOAD"";";     ! s11 class "C"
890  OUTPUT @Ana;"LABES22A""SHORT"";";          ! s22 class "A"
900  OUTPUT @Ana;"LABES22B""1/4 SHORT"";";      ! s22 class "B"
910  OUTPUT @Ana;"LABES22C""FIXED LOAD"";";     ! s22 class "C"
920  OUTPUT @Ana;"LABEFWDT""THRU"";";           ! fwd trans.
930  OUTPUT @Ana;"LABEFWDM""THRU"";";           ! fwd match
940  OUTPUT @Ana;"LABEREVT""THRU"";";           ! rev trans.
950  OUTPUT @Ana;"LABEREVM""THRU"";";           ! rev match
960  ! Label kit
970  OUTPUT @Ana;"LABK""WR90 A.0"";";
980  ! Done with kit; save into nonvolatile mem
```

```
990   OUTPUT @Ana;"KITD;SAVEUSEK;CALKUSED;"
1000  ! Set up analyzer for waveguide freq range
1010  OUTPUT @Ana;"STAR8.2GHZ;STOP12.4GHZ;"
1011  OUTPUT @Ana;"TITF1""WR90A0"""
1012  OUTPUT @Ana;"WAVD",Minf;"HZ;"
1020  END
```

Figure 4-4. Sample Program: Creating a Waveguide Calibration Kit

Program explanation

Line 100	Set minimum frequency (waveguide cut-off frequency).
Line 110	Set the maximum frequency for rectangular waveguide TE10 mode.
Line 120	Preset the network analyzer and set velocity factor for dry air.
Line 130	Set system impedance so that measurements are normalized to 1 ohm.
Line 140 through 250	Define Standard #1 as a "short" with zero delay and label as "SHORT."
Line 260 through 360	Define Standard #2 as a "short" with 32.633 picosecond delay and label as "1/4 SHORT" (1/4 wavelength offset delay).
Line 360 through 480	Define Standard #3 as a "load" with zero delay and label as "FIXED."
Line 490 through 590	Define Standard #4 as a "delay/thru" with zero delay and label as "THRU."
Line 600 through 700	Define Standard #5 as a "delay/thru" with 32.633 picosecond delay and label as "1/4 DELAY" (1/4 wavelength offset delay).
Line 710 through 840	Specify the standard numbers to be measured under a specific class.
Line 850 through 950	Assign a label for a specific class.
Line 960 and 970	Label the cal kit definition.
Line 980 and 990	Save the cal kit definition into the USER KIT and select USER KIT.
Line 1000 and 1010	Set up the nominal frequency range for X band waveguide.
Line 1011	Title file position 1 with "WR90A0" so that a subsequent STORE TO DISK will use this file name.
Line 1012	Select waveguide delay and specify the waveguide cut-off frequency for the delay calculations.

A

Reading Binary Files

The network analyzer can store data files in ASCII or binary on a LIF formatted disk. The HP 9000 Series 200 or 300 computer can read a LIF formatted disk. The following program reads a network analyzer binary file into a data array.

```
10    !
20    ! Reading Binary Files
30    !
40    ABORT 7
50    A$=""
60    INPUT "File name? ",A$
70    INPUT "Number of points? ",X
80    ALLOCATE Dat(1:X,1:2)
90    ASSIGN @Disk TO A$&":,700,0"; FORMAT OFF
100   ENTER @Disk; Dat(*)
110   PRINT Dat(*)
120   END
```

Figure A-1. Sample Program: Reading Binary Files

Program explanation

Line 60 through 70	Input file name and number of points.
Line 80	Allocate a data array.
Line 90 through 100	Read file into data array.
Line 110	Display data array.